Quiz 10:

Question: What are the differences between a client and a server?

- A **client starts communication** by requesting connection to a server.

- **Only a server can terminate an established connection**.

- There is **no difference between client and server after connection** has been established.

Question: Select all valid statements about names and addresses

- **Mapping of names to addresses is stored in the Domain Name System** (DNS), a distributed database that can be **contacted through DNS servers.**

- URLs (e.g. http://google.com) contain names

- Names provide easier-to-remember ways to reach networking interfaces

Question: What does DHCP mean and what can you use it for?

- DHCP can be used to **request and obtain an IP address**, from the available pool of IP addresses at the DHCP server, **so that the requesting device has a valid connection to the Internet**.

- DHCP can be used to **obtain a default gateway**, so that the requesting device knows where to send packets, when there no specific routes match the destination of these packets.
- DHCP stands for **Dynamic Host Configuration Protocol**.
- DHCP can be used to **obtain the address of one or more valid DNS server(s)**. This way, the requesting device can contact these servers to translate names into IP addresses.

Question: IP addresses are numbers (e.g. 128.104.125.20 or fe80::1c3c:a2d4:600a:e2df). What do these numbers identify?

1) A point of attachment in the Internet, i.e. the logical "position" of a networking interface in the network, like a sort of GPS coordinates for the Internet.

Quiz 20:

I'm lazy to do this one since it's just programming in C

Quiz 30:

Question: Order the sequence of steps in the lifetime of a (TCP) connection:

1) The server **listens** to a port (function listen() in C, constructor ServerSocket() in Java)
2) The client **connects** to the server on a particular port (function connect() in C, constructor Socket in Java)
3) The server **accepts** client's connection (accept() in Java and C)
4) Client and server negotiate the connection parameters in the hand-shake

5) **Transit** data back and forth (read and write operations in the client and server
6) Connection is **closed** (close() in C and Java)

Question: Match each property/characteristic with the right layer/protocol.

Provides ordering, i.e., bytes are received in the same order as they were transmitted.	Transport layer/ TCP
Connection-oriented, i.e. the connection has to be established and negotiated between both participants (through a hand-shake) before starting data exchange.	Transport layer/ TCP
Connectionless, i.e. no hand-shake is needed to exchange data.	Transport layer/ UDP
Provides reliability, i.e., transmitted bytes are acknowledged upon reception, retransmitted otherwise.	Transport layer/ TCP
Best-effort: packets can be reordered and lost.	Network layer/ IPv4 and IPv6
Provides integrity, i.e., the receiver has means to check whether received data is corrupted (for instance, by way of checksums).	Transport layer/ TCP and UDP

Question: Which of the following statements are correct? (Cast types)

- **Multicast** corresponds to communication between **one sender, and a set of recipients**. Recipients are named, but "unknown" to the sender.

- **Unicast** and anycast describe communication between **exactly one sender, and exactly one recipient**. In unicast, sender and recipient are uniquely identified. In anycast, the sender is uniquely identified, but the recipient is chosen from an unknown set of candidates.

Question: Select all correct statements about "Socket Communications."

- A **socket** is a **communication endpoint**.
- A **connection** is **uniquely identified by a pair of sockets**.
- **Multiple (different) connections** can **exist between the same pair** of networking interfaces (hosts).
- A **socket** is **described** as a **tuple (IPaddr,port)**, where IPaddr is a network (L3, e.g. IP) address of an interface and the port identifies the process that handles the communication in the corresponding device.

Question: What is a socket?

- A pair **(IP address, port)**
- An **endpoint** in a network communication

<u>Question:</u> In the class, two transport protocols were mentioned: TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). TCP provides reliable transport of packets, meaning, among other things, that its provides retransmissions if packets are lost, and ensures that data arrives to the destination in the same order in which they were sent by the source. UDP, in contrast, does not provide any of these services, meaning that packets send through UDP may be lost, or received in the wrong order; UDP does not allow the endpoints to notice whether this happens.

In these conditions, why would anyone use UDP instead of TCP?

1) In some applications, **reordering or loss of some packets is not critical -- and/or, can be handled through mechanisms at the application layer itself**. An example of this is live video streaming, in which loss of a particular video frame is less important than on-time reception of frames that actually arrive to destination. TCP provides reliability, but this comes at a cost (connection establishment, receiver acknowledgements, retransmissions, etc., in particular leading to possibly higher latency) that is not worth in some cases.

<u>Question:</u> In the class, two transport protocols were mentioned: TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). Other transport protocols exist: SCTP, DCCP, etc. In the TCP/IP model, the transport layer is responsible for maintaining end-to-end communication between two devices in a network.

Is it possible for a distributed application to send information (data packets), from a source to a destination, through a TCP/IP network without using *any* transport protocol?

1) True

Quiz 31:

Question: Consider the following C-code, and select from among the statements below all those which are true:

```
int socket_desc = socket(AF_INET6, SOCK_DGRAM, 0);
```

1) This code **creates** a **socket** using I**Pv6.**
2) This code can be **used** when writing a **client.**
3) This code can be **used** when writing a **server.**
4) This code **creates** a **UDP socket.**

Question: Consider the following code, which is part of a TCP server program:

```
int r = listen(s, b);
```

The first argument, s to the function listen() is the socket, over which the server should listen for incoming connections. The second argument, b, is the size of the queue of clients that can "wait for" a connection.

True

Question: Consider the following line of code, which is part of a program using TCP:

```
int r = accept(sockid, &clientAddr, &addrLen);
```

The function accept(...) is used by a server to accept new connections. It is a blocking function call. The server must previously have used a bind() on sockid. Sockid is exclusively used for accepting new connections. **Once a new connection has been established, the return value from accept(...) -r- indicates the socket for use data transfer with the connected client.**

True

Question: **In the following code, when using TCP:**

```
int status = bind(sockid, &addrport, size);
```

The 2nd argument (&addrport) is a pointer to a struct sockaddr, into which the IP address and port number of the peer is written when bind(...) returns.

False: address port is just port where the socket on the server is bound to on the server and will be listened from.

Question:

Consider the following program:

```
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>
int main(void){
   char* msg = "Hello World\n";
   struct sockaddr_in serverAddr;
   serverAddr.sin_family = AF_INET;
   serverAddr.sin_port = htons(5000);
   serverAddr.sin_addr.s_addr = htonl(INADDR_ANY);
   int sockid = socket(AF_INET, SOCK_STREAM, 0);
   int b = bind(sockid, (struct sockaddr*)&serverAddr, sizeof(serverAddr));
   int r = listen(sockid,10);
   int s = accept(sockid,(struct sockaddr*)NULL ,NULL);
   int t = send(s, msg, strlen(msg), 0);
   printf("sent %d bytes out of %lu",t,strlen(msg));
   return 0;
}
```

Select from among the statements below, those which are correct.

1) This program **compiles without errors.**
2) This is a **TCP Server**

Question: Consider this code, which is part of a TCP program:

```
int r = connect(sockid, &peerAddr, addrlen);
```

This code is used by a client, and the return value of this function indicates if connection was successful (0) or not (-1)

True

Quiz 32:

Question: Convert 32 bit integer fro network byte order to host byte order

To convert a 32 bit integer from network byte order to host byte order, you use

ntohl() ✔ function.

Question: Populate address

Assuming that the command line argument given when executing this program is an IP address of the format 130.225.194.2, select the right function that that would correctly populate `addr`:

```
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    struct in_addr addr;
    inet_pton
}
```

Question: UDP echo server order:

When writing an UDP Echo Server, the order of C networking primitives used will be:

socket(...)	✔	->
bind(...)	✔	->
recvfrom(...)	✔	->
sendto(...)	✔	->
Nothing -- we're done!	✔	->
Nothing -- we're done!	✔	->
Nothing -- we're done!	✔	->
Nothing -- we're done!	✔	

Question: When writing UDP, the function **send(...):**

When writing programs communicating using UDP in C, the `send(...)` function call can be

Used in neither the client nor the server. ✔ .

Quiz 40:

Question: What **flags** shall be **passed to the compiler,** while **linking** a program using posix threads?

-lpthread

Question: Among the arguments passed to pthread API **pthread_create(...)** the **final argument**, is:

Data being **passed** to the **thread when started**.

Question: The **datatype** used for **storing** a **thread id** is:

pthread_t

Quiz 50:

Question: In a protocol stack, a layer provides services to the immediately upper layer, while calls on services from the immediately lower layer.

Question: When the network layer decides to which transport layer protocol an incoming message is to be delivered, the header field protocol number in the network layer header is inspected.

Question: The set of identifiers {Ethertype, IP address, port number} serve to uniquely identify the path "from the network interface to the application" of an incoming packet.

False:

Question: Identifiers are carried in packet headers.

True

Question: As an incoming packet progresses upwards in the protocol stack from layer n to layer n+1, the packet header for layer n is processed and removed.

(Each layer resolves its own headers)

Quiz 60:

Question: DNS can be used for:

- Address to name translation.
- Alias resolution.
- Resource sharing.
- Anything that involves matching names and records
- Load-balancing.
- Name to address translation

Quiz 61:

Question: In the DNS namespace tree, a FQDN is:

a leaf in the tree, and all the upper nodes towards the root, with traversed edges corresponding to dots (.) connecting node labels (e.g. www.polytechnique.edu).

Question: In the DNS namespace tree, a FQDN is:

The **Zone File in** an **authoritative DNS server, contains resource records for all the names in a given domain.**

For example**, the zone file for the DNS server to whom example.com is delegated, contains resource records** for all names such **as www.example.com., www.office.example.com., and mail.office.example.com.**

True

Question: **In the DNS name space, a fully qualified domain name is a sequence of labels, separated by Dot.**

Question: **Select all formally valid Fully Qualified Domain Names (FQDNs).**

Select all formally valid Fully Qualified Domain Names (FQDNs).

Select one or more:

- [ ] a. polytechnique
- [ ] b. edu
- [ ] c. polytechnique.edu
- [x] d. www.polytechnique.edu. ✔
- [x] e. mail.polytechnique.edu. ✔
- [ ] f. http://www.polytechnique.edu
- [ ] g. http://www.polytechnique.edu:80
- [x] h. www.lix.polytechnique.edu. ✔

Quiz 62:

Question: **Match terms with definitions.**

- Recursive DNS Request: The servers to which a request is sent will either answer the request, if such an answer is available at its local cache, or query other DNS servers in order to get an answer that can be sent back to the requester

Question: A(n) recursive resolver returns a final answer to a DNS lookup (or query) for a name, regardless of if the name is within its own zone or not.

Question: The DNS server software on your computer, which communicates with the DNS system, is called an "Iterative Resolver".

False

Question: The DNS server, which your computer contacts directly for resolving a name, is a "Recursive Resolver".

True

Quiz 63:

Question: In DNS, several names, www.example.com, mail.example.com, and login.example.com, can map to resource records (A-records) with the same IP address.

True: **Note that the resource records are unique and different but the IP addresses stored as values are the same**

Quiz 64:

Question: In the DNS system, root servers…

- know (or, knows someone who knows) who is maintaining records for any given domain
- are hundreds of physical servers, that replicate 13 logical servers or "named authorities" ("a" to "m")
- must be synchronized with each other

Question: Select all correct sentences about authoritative servers.

- An authoritative name server for a domain contains a complete copy of the domain's information, and in particular, contains a SOA (Start Of Authority) record for this domain.

- The authoritative name server for a domain name provides answers to DNS queries involving this domain.
- An authoritative name server for a domain is a server able to provide original, non-cached answers to DNS queries about this domain.
-

Quiz 71:

Question: When a packet is received at the networking layer for a given destination, it is…

- sent through the route in the routing table with “longest match prefix” to the packet destination.
- sent through the host route in the routing table for which the destination is exactly the packet destination, if one exists.

Question: What does the "cost" represent in host routing tables?

- It is a unitless metric
- Its value is arbitrary, only useful in a host to distinguish between (1) the host itself, (2) local destinations (i.e. in the host link), (3) the rest of the Internet.
- **Note: the cost is most often economic instead of scientific, maybe routers have preferences to direct one way than another (the value is arbitrary)**

Quiz 80:

Question: In an Ethernet network (a cable), two stations transmit a packet at the same time, thus causing a collision on the link to which they are connected. Select all the valid statements on collision detection and avoidance below.

- If the minimal duration of a transmission is smaller than twice the maximum propagation time in the link, there are collisions that may be unnoticed.

- If a collision is detected by the transmitting stations, the involved stations retransmit after a random amount time delay. If a new collision occurs, a new retransmission is scheduled, with a time delay determined randomly over a longer interval. This interval for retransmission doubles each time a collision is detected, thus reducing the probability of new collisions -- this retransmission mechanism is denominated "exponential backoff".
- Neither of the two stations will detect a collision between their transmissions if, for instance, transmissions are too small -- or stations are "too far" (i.e., the propagation delay between them is too large).